

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleksander Avberšek

**Mobilna aplikacija za podporo
enostavnim projektom**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Organizacije, ki se ukvarjajo z organizacijo več vzporednih dogodkov, v bistvu dnevno izvajajo nekaj enostavnih projektov vzporedno. Pri tem je pomembno, da imajo vodje možnost razdeljevanja nalog v vsakem trenutku, izvajalci nalog pa možnost v vsakem trenutku pregledati seznam zadolžitev in označiti naloge, ki so jih že opravili. Zasnujte mobilno aplikacijo, ki informacijsko podpira izvajanje tovrstnih enostavnih projektov. Pri tem upoštevajte, da v organizaciji obstaja centralni sistem urnikov, ki naj ga mobilna aplikacija uporablja. Aplikacijo razvijte na platformi Windows Phone.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Aleksander Avberšek, z vpisno številko **24950299**, sem avtor diplomskega dela z naslovom:

Mobilna aplikacija za podporo enostavnim projektom

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 12. junij 2014

Podpis avtorja:

Iskreno se zahvaljujem mentorju doc. dr. Roku Rupniku, staršem in moji družini za vzpodbudo in čas.

Mojim dragim trem ...

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Splošno o sistemu urnikov v CD	3
3	Tehnologije, razvojna in programska orodja	7
3.1	Windows Phone	8
3.2	Visual Studio	11
3.3	C# in .NET ogrodje	13
3.4	Silverlight	14
3.5	XAML	14
3.6	Strežniška programska oprema	15
4	Aplikacija	17
4.1	Dostop do podatkov	17
4.2	Začetni zaslon	23
4.3	Urniki	25
4.4	Projekti	28
4.5	Zaposleni	31
4.6	Nastavitve	33
4.7	Uporabljeni dodatki, knjižnice in posebna orodja	34
4.8	Testiranje aplikacije	35

KAZALO

4.9	Distribucija aplikacije	38
5	Ideje za nadaljevanje	43

Seznam uporabljenih kratic

kratica	pomen
WP	Windows Phone
XML	eXtensible Markup Language
JS	javaScript
SDK	Software Developers Kit
SSIS	SQL Server Integration Services
MSDN	Microsoft Developers Network
XAML	eXtensible Application Markup Language
IIS	Internet Information Services
UI	User Interface
IDE	Integrated Development Environment
CIL	Common Intermediate Language
CLR	Common Language Runtime
RIA	Rich Internet Application
URL	Uniform Resource Locator
AET	Application Enrollment Token

Povzetek

V diplomski nalogi sem prikazal postopek izdelave mobilne aplikacije, ki prikazuje urnike dogajanja, podatke o projektih oz. prireditvah ter podatke o sodelujočih.

Razvoj mobilnih telefonov, ki je njihovo osnovno funkcijo »telefoniranja« razširil na zelo širok spekter storitev, nam omogoča dostop do raznovrstnih informacij praktično na vsakem koraku, od tukaj pa izvira ideja, da so zaposlenemu oz. udeležencu v procesu organizacije in izvedbe določene prireditve/projekta v CD, dostopni vsi podatki, ki so potrebni za brezhibno in pravočasno dokončanje njegovih zadolžitev. Pri samem procesu izvedbe prireditve/projekta je prisotnih več različnih aktivnosti, ki se med sabo ali prepletajo ali pa sledijo ena drugi, kar pomeni, da je časovna usklajenost bistvenega pomena.

Aplikacija naj bi uporabniku postregla s podatki o njegovem osebnem urniku in zadolžitvah ter s podatki o urnikih in zadolžitvah ostalih sodelujočih. Omogočen je tudi pregled nad celotno prireditvijo oz. projektom, kar uporabnikom omogoča še dodaten pregled nad dogajanjem. Z aplikacijo dobimo tudi pregled nad trenutno prisotnimi sodelujočimi na projektu in njihovimi kontaktnimi podatki.

Aplikacija je razvita za mobilno platformo Windows Phone, uporabili pa smo že obstoječ sistem planiranja urnikov v organizaciji.

Ključne besede: Windows Phone, mobilne aplikacije .

Abstract

In the thesis I describe the development of mobile application for providing work schedules, data about whole project and co-workers data.

Evolution of mobile phones has expand their primar task of phone calls to a wide range of services. The idea of an application is to provide every participant in the proccess of organization all information that is needed for flawless and accurate completing of tasks. The whole proccess consists of many activities which interact between each other and that is why the timing is crucial.

Mobile application should provide us information about our personal schedule of tasks and activities and also with the information about our co-workers schedules. There is also an overview for the whole project and all the participants.

The application is developed for Windows Phone OS and for data source we will use the existing system for planing activities and tasks.

Keywords: Windows Phone, mobile apps, computer.

Poglavje 1

Uvod

Vse večja razširjenost pametnih telefonov in njihova vse večja zmogljivost in enakost osebnemu računalniku, narekujeta tudi razvoj mobilnih aplikacij, ki uporabnikom na prilagojen način omogočajo dostop do informacij, ki so običajno bile dostopne samo preko namenskih aplikacij na osebnih računalnikih. Zanimiv je sam razvoj t.i. pametnih telefonov, ki skozi razvoj prevzemajo funkcionalnosti vse več naprav (od radijskih sprejemnikov, MP3 predvajalnikov, fotoaparátov, dlančnikov do navigacijskih naprav in v zadnjem času še tabličnih računalnikov).

V nadaljevanju predstavljena rešitev vsakemu sodelujočemu pri izvedbi določene prireditve oz. projekta omogoči pregled nad njegovim osebnim urnikom, ki mu ga določi njegov vodja, in pregled nad urniki ostalih sodelujočih.

Za osnovo in vir podatkov sem si izbral sistem za planiranje prireditev v Cankarjevem domu. V Cankarjevem domu se vsako leto zvrsti skoraj 2000 prireditev [1], od katerih vsaka zahteva natančno planiranje razpoložljivih resursov. Vsaka takšna prireditev je sestavljena iz množice urnikov, katerih ključni podatki so čas, prostor, aktivnost in sodelujoči zaposleni.

Aplikacija je razvita za mobilne aparate z operacijskim sistemom Windows Phone in deluje na omenjenem operacijskem sistemu od verzije 7.5 naprej.

Sama zgradba diplomske naloge (glavni del oz. analiza) je osnovana na

predstavitvi uporabe aplikacije, v kar pa dodajam podrobnejše opise gradnikov ali programskih rešitev. Te so predstavljene, ko se prvič pojavijo v opisu, ob morebitnem naslednjem pojavljanju pa je dodana referenca na mesto podrobnejšega opisa.

Poglavje 2

Splošno o sistemu urnikov v CD

Sistem, katerega eden izmed končnih rezultatov je tudi izpis urnika na mobilnem aparatu, je zasnovan na način, ki omogoča, da vodja projekta po predhodni rezervaciji in potrditvi le-te s strani programskega koordinatorja določi urnike izvajanja aktivnosti na tem projektu, tem urnikom pa vodje posameznik oddelkov določijo zaposlene, ki so zadolženi za njihovo realizacijo. Celoten postopek je sestavljen iz naslednjih korakov:

- (1) oddelek za marketing vzpostavi kontakt z morebitno stranko (ali obratno)
- (2) glede na strankine želje se preveri razpoložljivost resursov (terminov, lokacij, zaposlenih) in se okvirne termine zabeleži v sistemu urnikov
- (3) po sprejetju končne odločitve s strani stranke se projekt ali potrdi ali pa stornira (projekt se lahko stornira kadarkoli tekom priprave!)
- (4) določijo in vnesejo v sistem se okvirni urniki posameznih aktivnosti (npr. vaja orkestra, priprava luči, prihod udeležencev, otvoritev kongresa, itd...) in vodja projekta pripravi dispozicijo (v njej so točno definirane zahteve naročnika)

Št. oseb: 11

ODDELEK ZA ODRSKO-SCENSKE POSTAVITVE (2.6. - 8.6. 2014)							
Osebe Dnevi □ / □	PON - 2.6.2014 □	TOR - 3.6.2014 □	SRE - 4.6.2014 □	ČET - 5.6.2014 □	PET - 6.6.2014 □	SOB - 7.6.2014 □	NED - 8.6.2014
...
...
...
...
...

Slika 2.1: Primer prikaza urnikov v spletni aplikaciji

- (5) v kolikor imajo vodje posameznih področij izvedbe na voljo potrebne resurse potrdijo dispozicijo (oziroma so potrebna dodatna usklajevanja)
- (6) dispozicija je potrjena, ko se z njeno vsebino strinjajo vodje vseh udeleženih oddelkov
- (7) vodje oddelkov določijo delavce za posamezno aktivnost oz. urnik

Celoten sistem je sestavljen iz več spletnih aplikacij, ki omogočajo vnos in ažuriranje podatkov ter vrsto izpisov po različnih kriterijih. Na sliki 2.1 je primer izpisa, ki za izbrani teden prikazuje aktivnosti zaposlenih.

Administracijo in celoten pregled nad sistemom ima koordinator prireditel, ki za ta namen uporablja Windows Forms aplikacijo. Uporaba aplikacij je mogoča ob predhodni registraciji v sistem, sama kasnejša uporaba pa je določena glede na vlogo in pravice posameznega uporabnika.

Poleg zgoraj omenjenega sistema urnikov, naša mobilna aplikacija črpa podatke še iz naslednjih sistemov:

- sistem za registracijo delovnega časa (tukaj dobimo podatke o posameznikovi prisotnosti na delovnem mestu)
- interni imenik zaposlenih, kjer so na voljo vsi kontaktni podatki zaposlenega + podatki o prostorih in dežurnih službah

- sistem za evidentiranje odsotnosti z delovnega mesta (dopusti)

Vsi ti sistemi predhodno niso bili povsem povezani, kar pa je za uspešno realizacijo mobilne aplikacije bilo bistveno. V ta namen je bilo potrebno na strežniškem nivoju pripraviti ustrezne popravke - predvsem vmesne tabele, ki so povezale različne podatke.

Poglavje 3

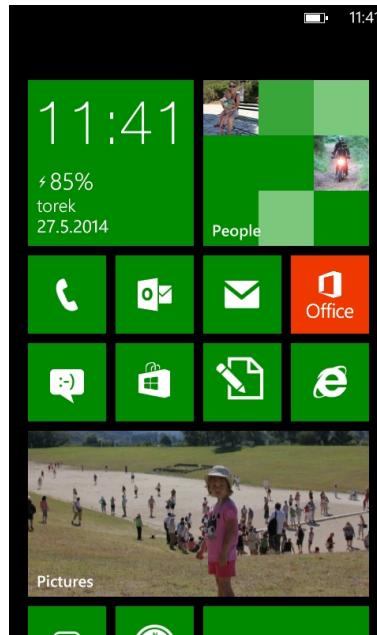
Tehnologije, razvojna in programska orodja

Za razvoj mobilne aplikacije smo uporabili razvojno orodje Microsoft Visual Studio 2012. Ikone smo oblikovali z orodjem Metro Studio 2.0 podjetja Syncfusion, manipulacije s podatkovno bazo pa smo izvajali z namenskim orodjem SQL Server Management Studio 2012.

Strežniški del aplikacije se nahaja na strežniku Microsoft Windows Server 2008 R2, pri čemer za podatkovni del skrbi podatkovni strežnik Microsoft SQL Server 2008 R2, za aplikacijski del pa spletni strežnik Internet Information Services 7.5 (IIS 7.5).

Predstavljena Windows Phone aplikacija je zgrajena s uporabniškim vmesnikom definiranim v XAML datotekah, dogodki in akcije pa so definirane v t.i. »*code-behind*« datotekah, ki pa so v našem primeru C# programska koda.

Uporabo te mobilne aplikacije bi radi omogočili tudi uporabnikom starejše verzije sistema Windows Phone, to je od verzije Windows Phone 7.5 naprej. Zaradi tega je razvoj mobilne aplikacije potekal z uporabo razširitve Windows Phone SDK 7.1 in Silverlight tehnologije.



Slika 3.1: Primer začetnega zaslona

3.1 Windows Phone

Windows Phone je operacijski sistem, ki ga razvija podjetje Microsoft. Zanj je značilen enostaven uporabniški vmesnik zasnovan po sistemu Modern UI (na začetku poimenovan Metro). Ploščice (tiles) s katerimi so predstavljene posamezne aplikacije so oblikovane preprosto in minimalistično. Uporabniški vmesnik v osnovi sestoji iz dveh zaslonov:

- začetni zaslon (Start screen)
- zaslon s seznamom vseh nameščenih aplikacij

Začetni zaslon sestoji iz ploščic, ki jih lahko poljubno razvrščamo ter jim spreminjamo velikost. Ploščice lahko tudi vsebujejo dodatne podatke (npr. število novih sporočil, ...), in se sproti posodablja (Live Tiles).

Aplikacije na napravo nameščamo preko t.i. Windows Phone Store (Trgovina Windows Phone), kjer se nahajajo tako plačljive, kot brezplačne aplikacije.

Še nekaj značilnih funkcionalnosti oz. aplikacij Windows Phone naprav:

- središče Office (Office Hub) - mobilna verzija zbirke pisarniških programov
- iskalnik Bing
- OneDrive - brezplačen prostor za shranjevanje podatkov v oblaku
- Skype

Prva verzija sistema z imenom Windows Phone 7 je bila javnosti predstavljena novembra leta 2010. Kmalu je sledil prvi večji popravek (maj 2011) z imenom Mango oz. Windows Phone 7.5, ki je sistemu dodal nekaj bistvenih izboljšav. Ta verzija operacijskega sistema je doživela še en večji popravek in sicer v januarju 2013 je bil izdan Windows Phone 7.8, ki je dodal nekaj lastnosti novega sistema Windows Phone 8.

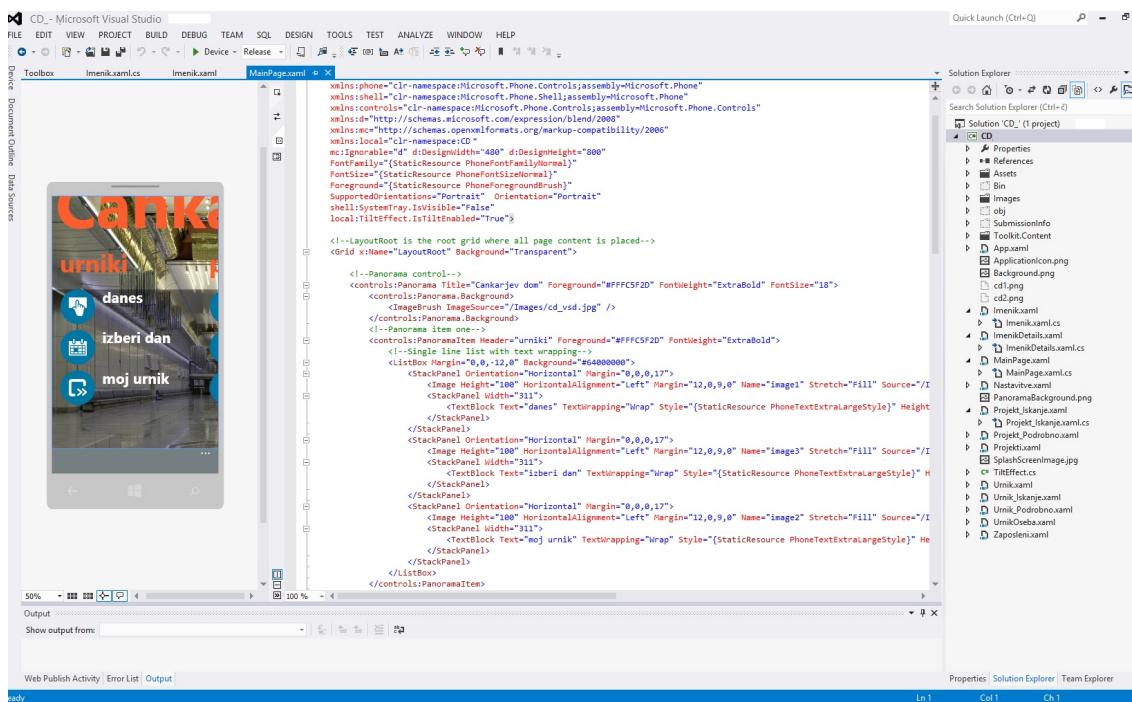
Windows Phone 8 je bil izdan v oktobru leta 2012 in je pomenil novo generacijo sistema. Nov sistem je osnovan na arhitekturi »*Windows NT kernel*«, kar pomeni, da starejših naprav z operacijskim sistemom WP7 ni mogoče nadgraditi na WP8, prav tako pa aplikacije napisane za WP8 ne delujejo na WP7, velja pa obratno in sicer lahko aplikacije za WP7 delujejo na WP8. Nova zasnova je operacijski sistem za telefone zelo približala sistemu za računalnike (Windows 8), saj si delijo precej gradnikov.[2] V aprilu 2014 je Microsoft izdal novo verzijo operacijskega sistema z imenom Windows Phone 8.1, ki prinaša precej novosti. Nekaj le-teh:

- tretji stolpec za ploščice na začetnem zaslonu
- mobilna verzija spletnega brskalnika Internet Explorer 11
- spremenjen Windows Phone Store
- izboljšan način vnosa preko tipkovnice t.i. »*WordFlow*«
- izboljšana večopravilnost

Strojne zahteve, ki so bile na začetku zelo strogo določene, se z novimi verzijami malo rahljajo, saj so opustili tudi zahteve glede obveznih gumbov na napravi. Zadnje minimalne zahteve (za Windows Phone 8.1) trenutno so:

- Qualcomm Snapdragon S4, Snapdragon 200, Snapdragon 400, ali Snapdragon 800 večjedrni procesor
- najmanj 512 Mb delovnega pomnilnika (1Gb za zaslone z ločljivostjo 720p/1080p)
- najmanj 4Gb flash pomnilnika
- GPS in A-GPS
- micro-USB 2.0 podpora
- kamera zadaj (vsaj VGA ločljivosti)
- merilnik pospeška
- WiFi 802.11b/g in Bluetooth
- vsaj 4-točkovni “*multi-touch*” kapacitivni zaslon

Statistični podatki glede uporabe Windows Phone na svetovni ravni letu ob koncu leta 2013 pripisujejo cca. 4 % delež med uporabniki pametnih telefonov [11]. Med temi uporabniki pa je po podatkih portala *AdDuplex* [10] iz meseca aprila 2014 trenutno, izmed uporabnikov Windows Phone operacijskega sistema, 78 % uporabnikov Windows Phone 8, skoraj 19 % pa še vedno Windows Phone 7 oz. ene izmed nadgradenj tega sistema.



Slika 3.2: Visual Studio 2012

3.2 Visual Studio

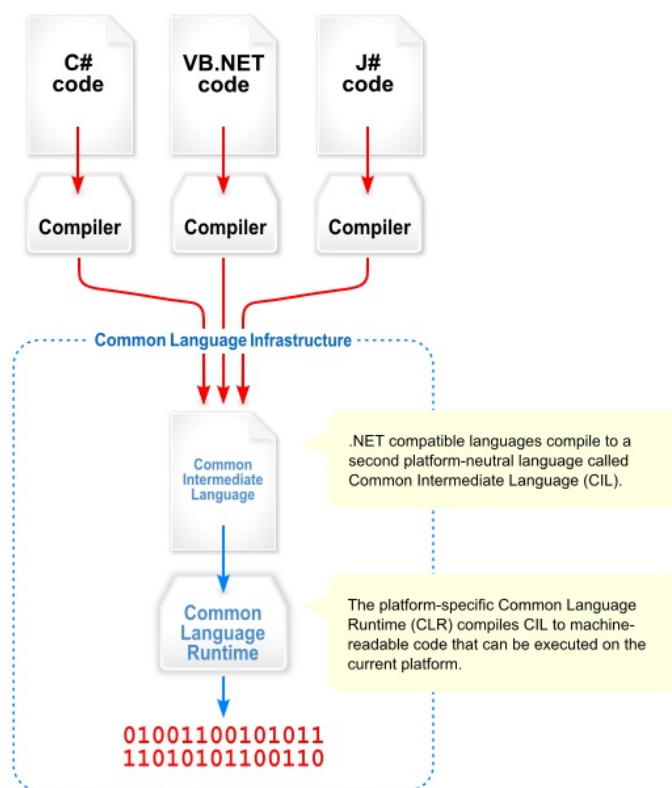
Razvojno orodje Visual Studio je integrirano razvojno orodje (IDE) podjetja Microsoft. Njegova uporaba obsega razvoj aplikacij za namizna in mobilna okolja do razvoja spletnih aplikacije ter servisov in še mnogo več. Vključuje uporabo različnih programskih jezikov, kot so C, C++, C#, Visual Basic, F#, preko dodatkov, pa je mogoča tudi uporaba jezikov kot so Python, Ruby in drugi [4].

Urejevalnik programske kode v Visual Studiu poskrbi za samodejno zaključevanje programske kode in vizualno poudari sintakso (IntelliSense). Razhroščevalnik, ki deluje tako na nivoju izvirne kode, kot na nivoju strojne kode pa poskrbi za lažje testiranje in odkrivanje napak. Za samo oblikovanje grafične podobe aplikacij imamo na voljo več tovrstnih urejevalnikov, kjer z metodo “*drag&drop*” oblikujemo oz. urejamo aplikacijo.

	Windows Phone 7	Windows Phone 8	Windows 8
programski jeziki	C#, Visual Basic, F# HTML5, JS	C#, Visual Basic C++/CX HTML5, JS	C# C++/CX HTML5, JS, WinJS
API (programski vmesnik)	.NET XNA	.NET DirectX WinRT	.NET DirectX WinRT

Tabela 3.1:

Kot smo omenili že zgoraj, se je s prehodom na novo verzijo Windows Phone 8 in s tem na novo arhitekturno zasnovo, spremenil tudi razvoj aplikacij za omenjene platforme. Tabela 3.1 prikazuje okvirni pregled po platformah, za katere razvijamo aplikacije [3]:

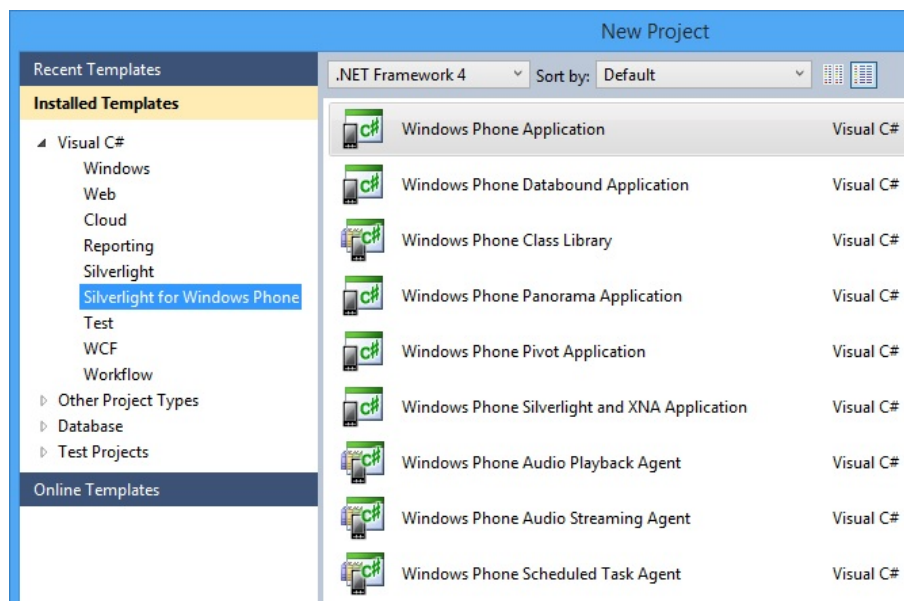


Slika 3.3: CLR [5]

3.3 C# in .NET ogrodje

Programski jezik, ki smo ga uporabili pri razvoju naše aplikacije je C#. Ta objektno usmerjen programski jezik je razvilo podjetje Microsoft na osnovah C++ za razvoj aplikacij z orodjem Visual Studio in .NET ogrodjem.

Prevajalnik našo programsko kodo (lahko je v kateremkoli jeziku, ki je združljiv z .NET) prevede v jezik, ki je neodvisen od platforme (CIL - Common Intermediate Language), nato pa se ta programska koda prevede v strojno kodo [5]. Za to poskrbi sistem imenovan CLR (Common Language Runtime), ki je odgovoren za izvajanje .NET programov[6]. Pod to odgovornost štejemo na primer upravljanje s pomnilnikom, upravljanje z izjemami, itd...



Slika 3.4: Silverlight za WP - izbira tipov aplikacij v Visual Studiu

3.4 Silverlight

Pri razvoju aplikacije smo se poslužili tudi aplikacijskega ogrodja Silverlight, ki ga je razvilo podjetje Microsoft [7]. Samo ogrodje je namenjeno razvoju obogatenih spletnih aplikacij (RIA - Rich Internet Applications), pri razvoju mobilnih aplikacij za Windows Phone pa se uporablja malo modificirana različica Silverlight 4, ki je tudi primarno okolje za razvoj aplikacij za Windows Phone 7.

3.5 XAML

XAML (eXtensible Application Markup Language) je označevalni jezik, ki se običajno uporablja za izdelavo uporabniških vmesnikov za aplikacije temelječe na .NET ogrodju. Poleg elementov uporabniškega vmesnika z njim definiramo tudi povezave med gradniki in podatki (t.i. data binding), dogodke in ostale stvari. Med drugim ga lahko uporabimo tudi za zapis vektorske

grafike in animacije.

XAML datoteke za oblikovanje uporabniškega vmesnika lahko urejamo z namenskim orodjem Microsoft Expression Blend, z Visual Studiem pa lahko urejamo, tako po metodi “*drag&drop*” (gradnike dodajamo in razmeščamo po zaslonu), kot s samim tekstovnim urejevalnikom, kjer nam je v pomoč tudi IntelliSense.

3.6 Strežniška programska oprema

Strežniški del aplikacije teče na strežniškem operacijskem sistemu Microsoft Windows Server 2008 R2, na katerem je nameščen podatkovni strežnik Microsoft SQL Server 2008 R2 in spletni strežnik IIS7.5 (Internet Information Services).

Za upravljanjem in delo s Microsoftovim podatkovnim strežnikom smo uporabili orodje Microsoft SQL Server Management 2012.

Poglavje 4

Aplikacija

Tukaj predstavljena aplikacija omogoča vsakemu zaposlenemu oz. sodelujočemu pri izvedbi določenega projekta pregled nad njegovim osebnim urnikom in urnikom ostalih sodelujočih ter vpogled v podatke o samem projektu. Aplikacija nam omogoča tudi dostop do kontaktnih podatkov sodelujočih.

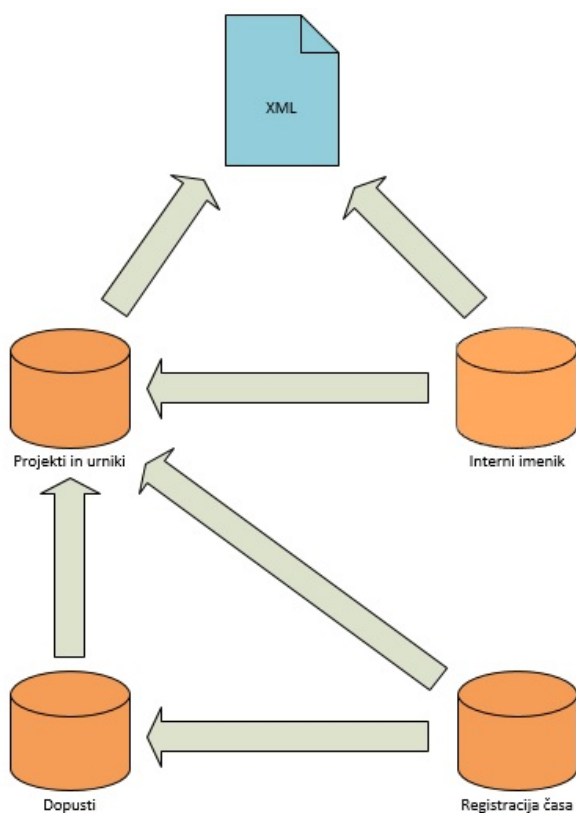
4.1 Dostop do podatkov

4.1.1 Podatkovna baza

Podatki se nahajajo v podatkovni bazi na strežniku in za dostop do njih je bilo potrebno narediti ustrezne storitve, ki nam te podatke posredujejo.

Za potrebe prikaza teh podatkov je bilo potrebno narediti spremembe na obstoječem podatkovnem modelu. Vsi podatki, ki se uporabljajo v naši aplikaciji, se dejansko nahajajo v štirih (4) različnih podatkovnih bazah ter na dveh različnih podatkovnih strežnikih:

- podatkovna baza (PB) sistema urnikov
- PB sistema za registracijo delovnega časa in PB sistema za evidenco dopustov
- PB internega imenika



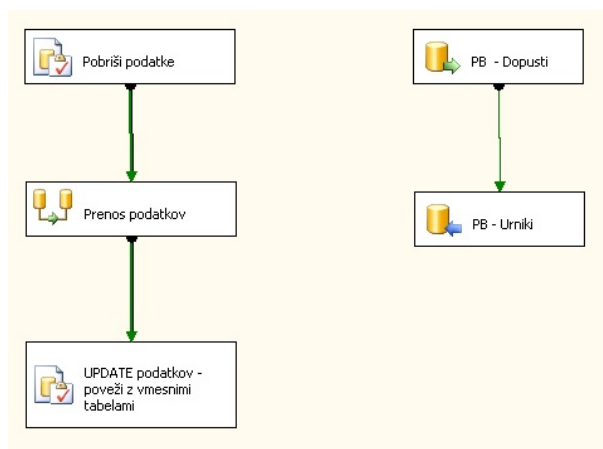
Slika 4.1: Podatkovni tok

Glavnina podatkov izhaja iz PB sistema urnikov, v katero se tudi prenašajo podatki iz ostalih PB. Prenos in kasnejša manipulacija s podatki je narejena s storitvijo SQL Server Integration Services (SSIS), ki je del podatkovnega strežnika Microsoft SQL Server.

SQL Server Integration Services (SSIS)

SSIS je platforma za integracijo podatkov in platforma za aplikacije za delovne tokove (workflow applications) [8]. Prvič se pojavi v verziji 2005 Microsoftovega SQL Serverja in s tem nadomesti Data Transformation Services (DTS).

Za uporabo omenjenih storitev je potrebno ustvariti SSIS paket (SSIS



Slika 4.2: Primer kontrolnega in podatkovnega toka v SSIS paketu

package), ki lahko vsebuje različne gradnike (povezave, naloge, ...), definicije kontrolnih in podatkovnih tokov ter definicije dogodkov. SSIS pakete smo kreirali z vizualnim orodjem, ki temelji na Visual Studiu 2008 in jih shranili v podatkovno bazo (pakete SSIS je možno shranjevati tudi v datoteke).

Podatkovna baza za celotno rešitev planiranja prireditve obsega preko 100 tabel, za potrebe naše aplikacije pa smo dodatno ustvarili še nekaj pogledov »View«, ki nam izberejo in prikažejo podatke, potrebne za prikaz v naši mobilni aplikaciji. Za povezavo z ostalimi potrebnimi bazami podatkov smo dodali vmesne tabele, v nekaterih primerih pa obstoječim tabelam dodali nova polja, ki vsebujejo tuj ključ.

Prenosi oz. usklajevanja podatkov potekajo v primeru relacije PB Urnik - PB Dopusti na podlagi prožilca (trigger), v primeru relacije PB Urnik - PB Reg. časa pa po vnaprej določenem urniku (schedule).

4.1.2 Spletne storitve

Podatke, ki jih prikazujemo na mobilni aplikaciji, dobimo iz podatkovnega strežnika preko aplikacijskega strežnika, za kar v našem primeru uporabljamo Internet Information Services 7.5 (IIS 7.5), ki je del strežniškega operacijskega

sistema Microsoft Windows Server 2008 R2.

Zahtevo za podatke posredujemo na spletni strežnik, morebitne kriterije pa dodamo kot parametre v URL naslovu. Po prejemu zahteve strežnik izvede ustrezno akcijo in vrne željene podatke. Podatki se na mobilno napravo prenesejo v obliki XML (eXtensible Markup Language oz. razširljiv označevalni jezik).

Seznam podatkov, ki jih lahko zahtevamo (možni parametri so zapisani v oklepaju):

- **DISPOZICIJE** (ID-Dispozicija, ID-Projekt, ID-Oseba, ID-Tip-Dispozicija)
vrne seznam dispozicij glede na podane parametre
- **DOPUST** (ID-Oseba, datum, od,do)
vrne seznam oseb, ki so na dopusti glede na podane parametre
- **IMENIK** (ID-Oseba, sektor, oddelek, skupina, naziv, delovno mesto, telefon, prostor)
vrne seznam kontaktnih podatkov za osebe glede na parametre
- **PRISOTNI** (ID-Oseba)
vrne seznam trenutno prisotnih oseb
- **PROJEKT** (ID-Projekt, od, do, naziv, vodja, status, koda)
vrne seznam projektov glede na podane parametre
- **URNIK** (ID-Urn timer, ID-Projekt, lokacija, od, do, vodja, aktivnost, koda)
vrne seznam urnik po željenih parametrih
- **URNIK-OSEBA** (ID-Oseba, ID-Projekt, ID-Urn timer, od, do)
vrne seznam urnikov za osebe (glede na parametre)

Tukaj je predstavljen preprost primer, ki nam kot odgovor na zahtevo vrne seznam prisotnih oseb (v XML obliki):

```
string kriteriji = "";
if (Request.QueryString["id"] != null)
    kriteriji = kriteriji + " AND(ID_TS=" + Request.QueryString["id"] + ") ";
String sConn = "connection string .....";
SqlConnection conn = new SqlConnection(sConn);
XmlDocument x = new XmlDocument();
XPathNavigator xpathnav = x.CreateNavigator();
conn.Open();
SqlCommand command = new SqlCommand("SELECT id_ts,id_oseba,naziv_oseba "+
    " FROM URN_Prisotni " +
    " WHERE (id_TS>0)AND(ID_Oseba IS NOT NULL)" + kriteriji +
    " ORDER BY naziv_oseba,id_TS " +
    " for XML RAW ('Prisotni'), ELEMENTS", conn);
using (XmlWriter xw = xpathnav.PrependChild())
{
    xw.WriteStartElement("Urnik");
    xw.WriteAttributeString("Avtor", "CD");
    xw.WriteAttributeString("Podatki", "Prisotni zaposleni");
    using (XmlReader xr = command.ExecuteXmlReader())
    {
        xw.WriteNode(xr, true);
    }
    xw.WriteEndElement();
}
conn.Close();
Response.ContentType = "text/xml";
Response.ClearContent();
x.Save(Response.Output);
Response.End();
```

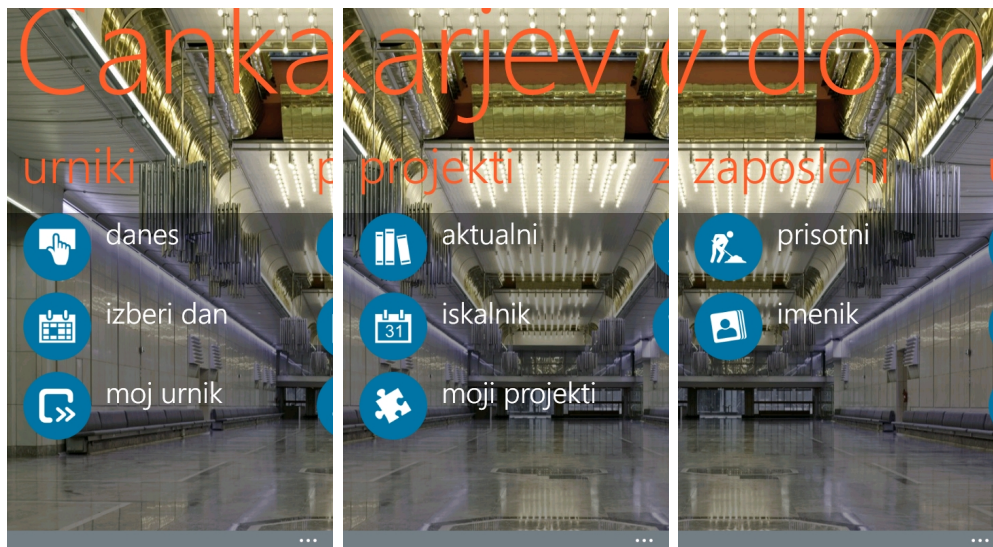
Po klicu zahteve za željene podatke moramo le-te še prebrati in jih ustrezno prikazati. V naši aplikaciji za to uporabimo razred *WebClient*, ki je del imenskega prostora *System.Net*. V naslednjem izseku programske kode vidimo potek:

```
WebClient wc_z = new WebClient();  
wc_z.OpenReadCompleted += wc_OpenReadCompleted_Prisotni;  
string uriString = "http://...naslov.../Prisotni.aspx";  
wc_z.OpenReadAsync(new Uri(uriString));
```

Kot je razvidno iz programske kode, se po končanju branja sproži dogodek *onOpenReadCompleted*, ki povzroči klic metode v kateri smo definirali naše podatke:

```
private void wc_OpenReadCompleted_Prisotni(object sender, EventArgs e)  
{  
    using (Stream s = e.Result)  
    {  
        XDocument data = XDocument.Load(s);  
        var Prisotni = from query in data.Descendants("Prisotni")  
                        select new Prisotni_Class  
                        {  
                            id_ts = (string)query.Element("id_ts"),  
                            id_oseba = (string)query.Element("id_oseba"),  
                            naziv_oseba = (string)query.Element("naziv_oseba")  
                        };  
        listBox1.ItemsSource = Prisotni;  
    }  
}
```

Pridobljene podatke v našem primeru izpišemo s pomočjo gradnika *ListBox*. Več o tem gradniku v [4.3.1], kjer je tudi primer predloge oblikovanega izpisa.



Slika 4.3: Začetni zaslon aplikacije

4.2 Začetni zaslon

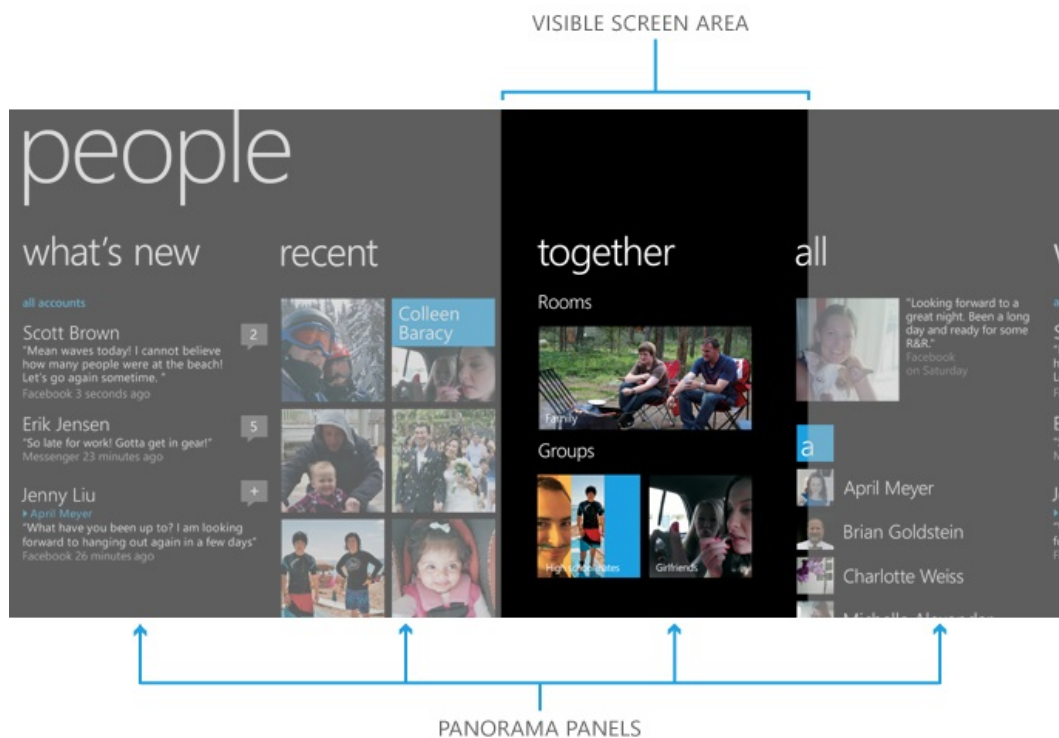
Za začetni zaslon smo izbrali gradnik, ki se imenuje »*Panorama*« in po katerem se med različnimi zasloni (menuji) premikamo s potegom po zaslonu na levo ali desno.

Osnovni menu je razdeljen na tri sklope:

- urniki
- projekti
- zaposleni

S pritiskom na posamezen gumb oz. napis znotraj posameznega sklopa, se nam odpre željeno področje.

Na dnu zaslona se nahaja aplikacijska vrstica (Application Bar oz. AppBar), v kateri je bližnjica do nastavitev za aplikacijo. Vsebino vrstice prikažemo s pritiskom na tri pikice.

Slika 4.4: Gradnik *Panorama* [9]

4.2.1 Gradnik *Panorama*

Panorama je standarden Windows Phone gradnik, ki nam omogoči, da naša aplikacija ni zgolj omejena na fizično velikost zaslona ampak nam ponuja poseben pogled, ki si ga najlažje predstavljamo kot široko platno, po katerem se premikamo levo-desno. Gradnik sestoji iz več kontrolnikov *PanoramaItem*, na katerih so nanizani ostali gradniki. Gradnik *Panorama* je možno prikazati samo v pokončnem (portrait) položaju.

4.3 Urniki

V razdelku »*urniki*« imamo dostop do pregleda urnikov na tri načine:

- današnji urnik (**danes**)
- urnik za poljuben dan, ki ga sami določimo (**izberi dan**)
- osebni urnik zaposlenega, ki je uporabnik aplikacije (**moj urnik**)

4.3.1 Prikaz seznama urnikov

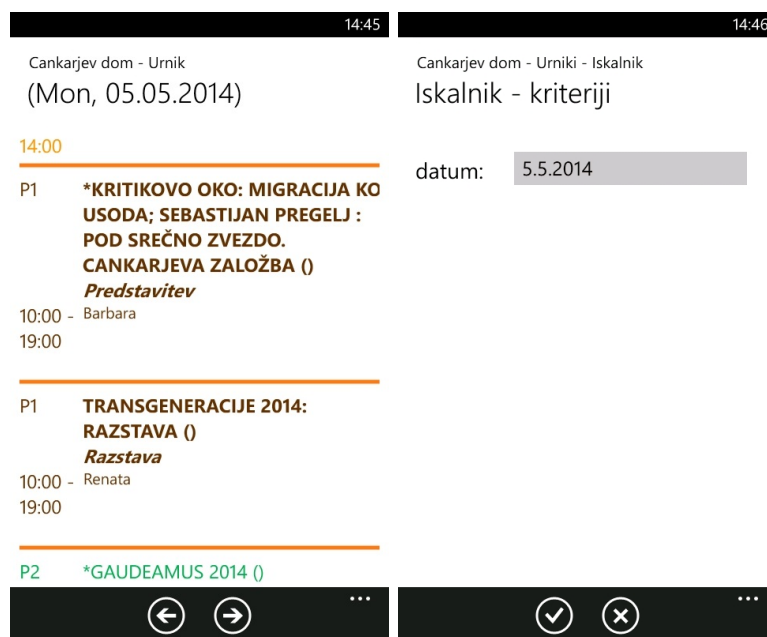
Po izbiri današnjega urnika ali določenega poljubnega dneva, se nam prikaže seznam urnikov za ta dan, kot je razvidno na sliki [4.5]. Iz seznama je razvidna lokacija, ura (od-do), naslov prireditve (projekta), aktivnost in kdo je vodja projekta. S puščicami v aplikacijski vrstici (AppBar) se pomikamo po dnevih naprej/nazaj. V kolikor smo se odločili za izbiro določenega dne (**izberi dan**) se nam pokaže urnik za ta dan.

Za prikaz seznama urnikov smo uporabili gradnik *ListBox*, ki smo ga napolnili z ustreznimi podatki.

Gradnik *ListBox*

Sezname različnih podatkov lahko prikazujemo z gradnikom *ListBox*. Ta gradnik nam omogoča vertikalni prikaz podatkov. Za prikaz podatkov vsebovanih v kakšni zbirki, moramo gradnik povezati s to zbirko (določimo lastnost *ItemsSource*). Posamezen zapis v celotnem seznamu oblikujemo s kreiranjem predloge.

V naslednjem izseku programske kode XAML smo posamezen zapis oblikovali v obliki petih vrstic (*TextBlock*), katerim smo z določitvijo lastnosti *Binding* določili polje iz podatkovne zbirke, katerih vrednosti se preslikajo sem. Takšen oblikovan zapis se ustvari za vsak zapis v zbirki podatkov.



Slika 4.5: Seznam urnikov in izbira dneva

```

<ListBox x:Name="listBox1" ItemsSource="{Binding}" Tap="listBox1_Tap">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel Margin="0,5,0,5">
        <Grid>
          <TextBlock Text="{Binding cela_lokacija}"/>
          <TextBlock Text="{Binding storno}"/>
          <TextBlock Text="{Binding ura}"/>
          <TextBlock Text="{Binding cel_naziv}"/>
          <TextBlock Text="{Binding naziv_aktivnost}"/>
        </Grid>
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>

```



Slika 4.6: Podrobno o urniku in seznam sodelujočih zaposlenih

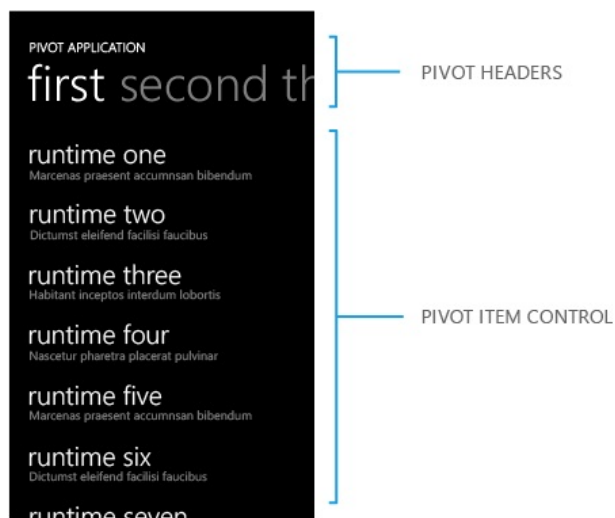
4.3.2 Podrobnosti o urniku

Z pritiskom na določen izbrani urnik v seznamu, se nam odpre nov zaslon s podrobnimi podatki o izbranem urniku in seznamom sodelujočih oseb. Tukaj smo uporabili gradnik *Pivot*, ki nam prikaže več zaslonov nanizanih eden poleg drugega.

Na tem zaslonu imamo v aplikacijski vrstici (AppBar) dve ikoni, ki nam ob aktiviranju izpišejo kontaktne podatke vodje projekta oz. izpišejo podrobnejše podatke o projektu.

Gradnik *Pivot*

Gradnik *Pivot* nam omogoča, da na hiter in enostaven način preklapljammo med različnimi zaslone/pogledi. Uporabimo ga lahko npr. za prikaz grupiranih podatkov, za prikaz različnih zaslonov, itd. . . Gradnik je sestavljen iz glave (pivot headers), kjer so izpisana imena zaslonov, in iz predmetov (pivot item), kjer je izpisana dejanska vsebina izbranega zaslona[12]

Slika 4.7: Zgradba gradnika *Pivot*

4.4 Projekti

Vsi urniki običajno izhajajo iz projektov oz. prireditvev. Poenostavljeno bi lahko rekli, da ni urnika brez predhodno definiranega in natančno opredeljenega projekta.

V razdelku "*projekti*" je urejen dostop do pregleda projektov na naslednje načine:

- trenutno odvijajoči se projekti (**aktualni**)
- iskalnik projektov po različnih kriterijih (**iskalnik**)
- projekti pri katerih je vodja uporabnik aplikacije (**moji projekti**)

Za prikaz aktualnih projektov je pomembno, da imajo definirane urnike, ki se trenutno (danes) izvajajo.

Naslednja možnost je uporaba iskalnika, ki nam omogoča iskanje projektov po naslednjih kriterijih:

14:48

Cankarjev dom - Projekti

aktualni (Mon, 05.05.2014)

406050214

Oblikovalska identiteta: Petra Veber

03.04.2014 - 07.05.2014

Potrjeno

Damjan

406040514

Zeitraum, Annibel Cunoldi Attems

15.04.2014 - 25.05.2014

Potrjeno

Damjan

642201414

Delavnica Wider

16.04.2014 - 15.05.2014

Potrjeno

Tina

405031114

Transgeneracije 2014: Razstava

17.04.2014 - 10.05.2014

Potrjeno

14:51

Cankarjev dom - Projekti - Iskalnik

Iskalnik - kriteriji

datum od:

5.5.2014

✓

datum do:

5.5.2014

■

koda:

■

vodja:

Avberšek Aleksanc

✓

naziv pr.:

■

status:

Potrjeno

✓

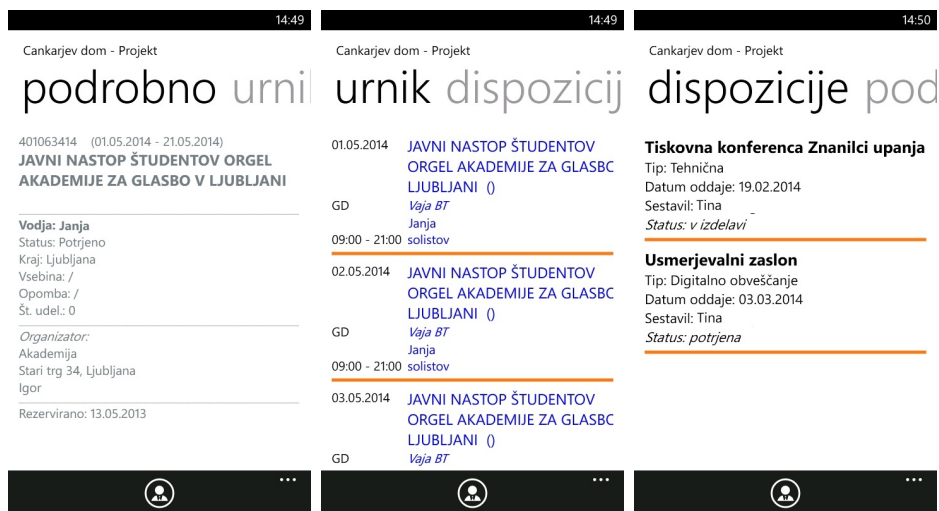
✓

✗

...

Slika 4.8: Seznam projektov in iskalni kriteriji pri projektih

- **datum od** – rezultat so projekti, ki so potekali (imajo določene urnike) od tega datuma naprej
- **datum do** – rezultat so projekti, ki so potekali do tega datuma
- **koda** – interna koda projekta
- **vodja** – vodja oz. odgovorni za projekt
- **naziv projekta** – iskanje po ključni besedi, če se nahaja v nazivu projekta
- **status** – pri projektih imamo lahko tri (3) statute projekta:
 - rezervacija** – projekt je v fazi dogovarjanja z morebitnim organizatorjem
 - potrjeno** – projekt je potrjen in se bo izvajal
 - storno** – projekt je preklican



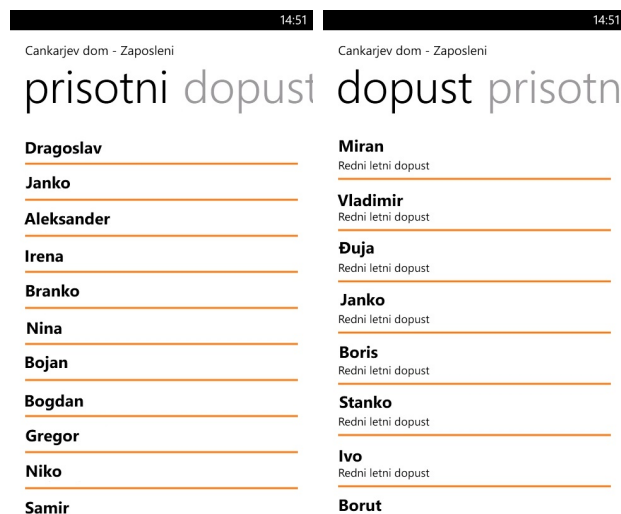
Slika 4.9: Podrobno o projektu

Zadnja možnost je izbira prikaza »*mojih projektov*«, ki nam vrne seznam vseh projektov, pri katerih smo vodja projekta.

Vsaka izmed izbir nam vrne seznam projektov, ki jih prikažemo z gradnikom *ListBox* (glej str. 25). Z izbiro posameznega projekta se nam izpišejo njegove podrobnosti v naslednjih predmetih gradnika *Pivot*:

- **podrobno** – izpiše najpomembnejše podatke o projektu (naziv, koda, termin izvajanja, vodja, status, kraj, organizator, itd. . .)
- **urnik** – izpiše celoten urnik za izbrani projekt, pri čemer je izpis podoben in ima podobne lastnosti, kot izpis urnikov opisan na strani 25.
- **dispozicije** – izpiše seznam dispozicij (podrobnih specifikacij aktivnosti in zadolžitve) za izbrani projekt. Izbira posamezne dispozicije nam le-to odpre v privzetem spletnem brskalniku, kjer si lahko ogledamo vse podrobnosti, vključno z morebitnimi priponkami.

Na tem zaslonu imamo v aplikacijski vrstici (AppBar) ikono, ki predstavlja bližnjico do kontaktnih podatkov vodje projekta.



Slika 4.10: Prikaz zaposlenih

4.5 Zaposleni

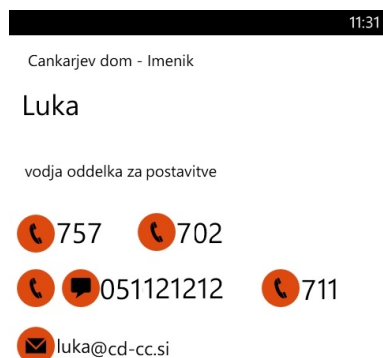
Pri izvedbi projektov so bistven dejavnik tudi ljudje, ki sodelujejo pri njegovi realizaciji. Na strani “*zaposleni*” imamo možnost dostopa do podatkov o zaposlenih (prisotni, na dopustu, kontaktni podatki, ...).

Prikaz “*prisotni*” nam v gradniku *Pivot* z dvema izbirama prikaže seznam trenutno prisotnih zaposlenih in prikaz zaposlenih, ki so trenutno celodnevno odsotni z dela.

Podatke o prisotnih zaposlenih dobimo iz sistema za registracijo delovnega časa. Prej omenjen sistem deluje na posebnem namenskem strežniku, zato je bilo potrebno urediti ustrezen prenos podatkov med strežnikoma (več o tem na str.18).

Prikaz “*dopust*” prikazuje osebe z najavljeno celodnevno odsotnostjo (dopust). Te podatke prav tako dobimo iz ločenega sistema.

V obeh zgoraj naštetih primerih dobimo izpis seznama oseb v gradniku *ListBox* (glej str. 25), kateri nam ob pritisku izpiše podrobne kontaktne podatke o izbrani osebi (slika 4.11).






Slika 4.11: Prikaz kontaktnih podatkov

Z izbiro prikaza imenika pridemo do zaslona tipa *Pivot*, kjer so podatki razvrščeni v tri sklope:

- **osebe** – seznam zaposlenih oseb z vsemi kontaktnimi podatki
- **prostori** – seznam prostorov, kjer se nahaja stacionarni telefon
- **dežurni** – seznam kontaktnih podatkov dežurnih služb

Vsi ti podatki se izpisujejo v gradniku *ListBox*, ob pritisku na izbranega pa se prikažejo podrobni kontaktni podatki (slika 4.11).

S pritiskom na ikono poleg podatka se lahko izvedejo naslednje akcije:

-  pokliče se izbrana številka
-  pošlje se kratko sporočilo (SMS)
-  pošlje se sporočilo po elektronski pošti

4.6 Nastavitve

Za pravilno delovanje potrebuje aplikacija nekatere podatke, ki jih vpišemo med nastavitve. Do nastavitve pridemo preko aplikacijske vrstice (AppBar) na začetnem zaslonu. Podatki, ki so potrebni za pravilno delovanje aplikacije so *uporabniško ime uporabnika* in *ID številka uporabnika*.

Nastavitve se v telefonu shranijo v t.i. *Isolated Storage*. Slabost tega načina je da se ob morebitni ponovni inštalaciji podatki izgubijo. Poleg tega načina shranjevanja podatkov, poznamo še nekaj načinov.

Shranjevanje podatkov v WP

Podatke lahko shranimo v [13]:

- Isolated Storage - je mesto v pomnilniku, ki je dostopno samo lastni aplikaciji. Vsaka aplikacija ima svoj *IsolatedStorage*. Vanj lahko shranjujemo tako datoteke kot mape, z uporabo razreda *IsolatedStorageSettings* pa lahko shranimo podatke v obliki para: ključ in vrednost (mi smo uporabili ta način)
- Remote Storage - podatki so shranjeni v oblaku ali drugje na spletu
- lokalna podatkovna baza

4.7 Uporabljeni dodatki, knjižnice in posebna orodja

Pri razvoju aplikacije smo uporabili tudi nekaj dodatnih orodij in knjižnic.

4.7.1 Windows Phone Toolkit

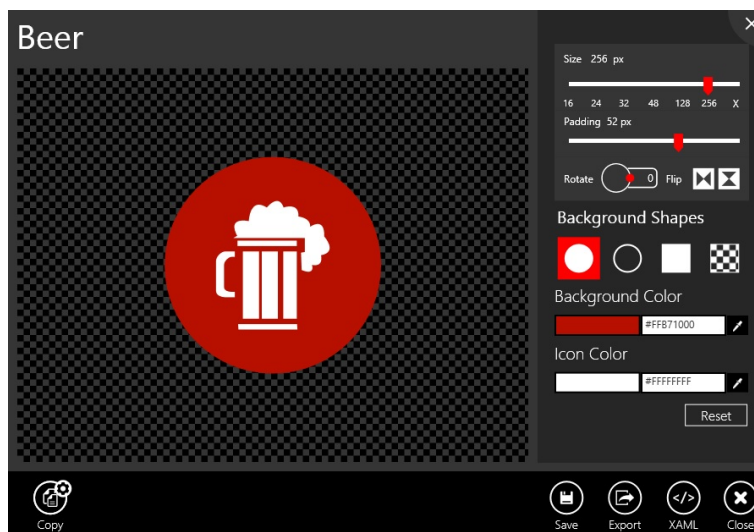
Windows Phone Toolkit je skupek dodatnih gradnikov in dodanih funkcionalnosti, ki nam olajšajo sam razvoj oz. nam vizualno polepšajo aplikacijo. Orodje je na voljo brezplačno na naslovu: <http://phone.codeplex.com>.

Izmed obsežnega nabora gradnikov smo pri razvoju naše aplikacije uporabili samo dva gradnika:

- TiltEffect - ostalim gradnikom (gumbi, seznam, ...) doda izgled oz. animacijo, kakor da so se ob pritisku rahlo pogreznili
- Long List Selector - malo naprednejši gradnik *ListBox*

4.7.2 Metro Studio 2.0

Metro Studio je izdelek podjetja Syncfusion, ki nam na zelo enostaven način omogoča ustvariti grafične podobe, ki nekako ustrezajo celotnemu konceptu Modern UI. Z njim smo ustvarili ikone, ki smo jih uporabili v aplikaciji.



Slika 4.12: Primer izdelave ikone s programom Metro Studio 2.0

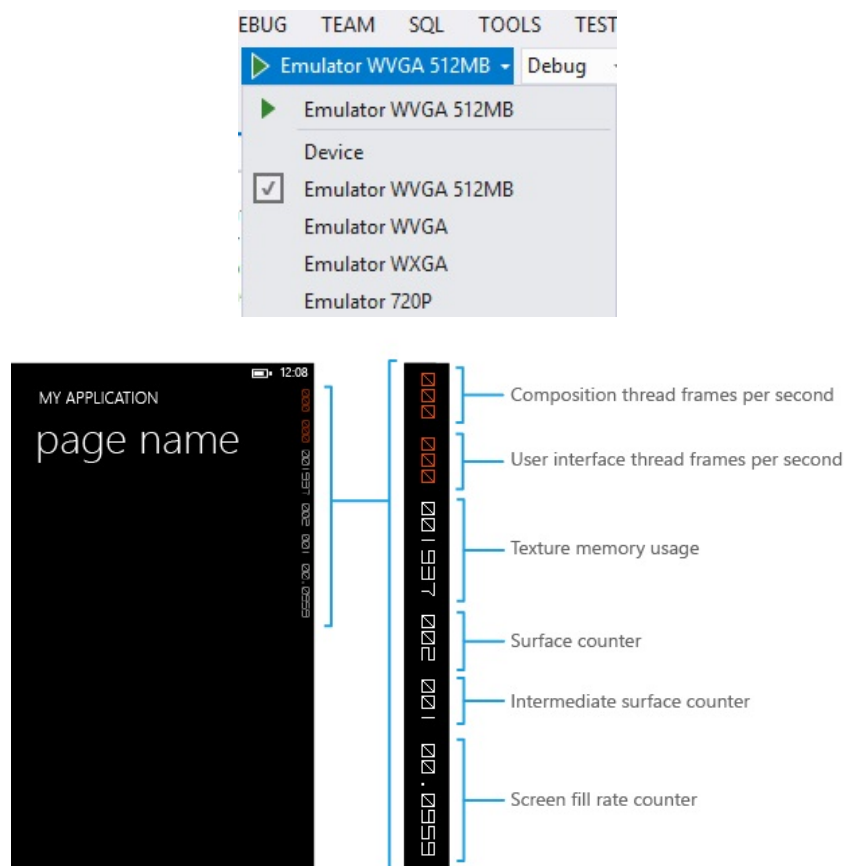
4.8 Testiranje aplikacije

Pri gradnji aplikacije je zelo pomembno tudi sprotno testiranje delovanja le-te. To lahko storimo na sami Windows Phone napravi, lahko pa delovanje aplikacije preverjamo tudi z uporabo Windows Phone emulatorja, ki posnema delovanje fizične naprave.

Windows Phone Emulator nam zagotovi virtualno okolje v katerem lahko testiramo delovanje WP aplikacij [14]. Emulator nam zagotovi tudi različne verzije sistemov, z različnimi ločljivostmi zaslonov s čimer nam je omogočeno preiskušanje aplikacije na več različnih napravah.

Upravljanje z emulatorjem je podobno, kot upravljanje fizične naprave. Gumbi, ki so na voljo na fizičnih napravah so simulirani s funkcijskimi tipkami:

- F1** - gumb *Nazaj*
- F2** - gumb *Windows*
- F3** - gumb *Iskanje*
- F6** - gumb *Kamera* - *pritisni do polovice*
- F7** - gumb *Kamera*
- F9** - gumb *Zmanjšaj glasnost (-)*



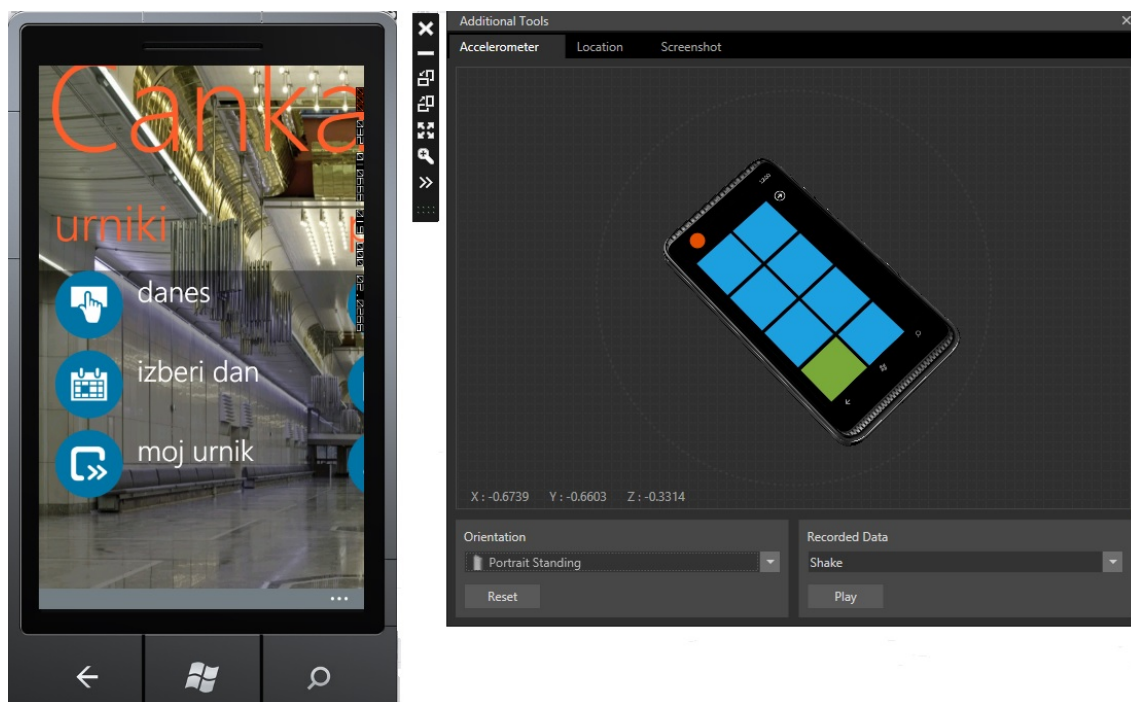
Slika 4.13: Izbira emulatorja in pomen povratnih informacij pri emulatorju

F10 - gumb *Povečaj glasnost(+)*

F12 - gumb *Vklop ali Zakleni zaslon*

Z Windows Phone emulatorjem lahko simuliramo tudi različne postavitev naprave. Za ta namen imamo poleg okna, v katerem se izvaja emulator, priročen menu v katerem lahko izberemo obračanje emulatorja v smeri urinega kazalca ali obratno. S tem simuliramo fizično obračanje naprave in testiramo obnašanje aplikacije pri različnih postavitvah (pokončno ali ležeče).

Poleg prej omenjenih funkcij ima Windows Phone emulator na voljo tudi več dodatnih orodij, s katerimi simuliramo uporabo nekaterih funkcionalnosti



Slika 4.14: Emulator, menu in dodatna orodja

same fizične naprave. Do teh dodatnih orodij pridemo preko menuja, dodatna orodja, ki so nam na voljo pa so:

- Merilnik pospeška (*accelerometer*)
- Lokacija (*location*)
- Posnetek zaslona (*screenshot*)

4.9 Distribucija aplikacije

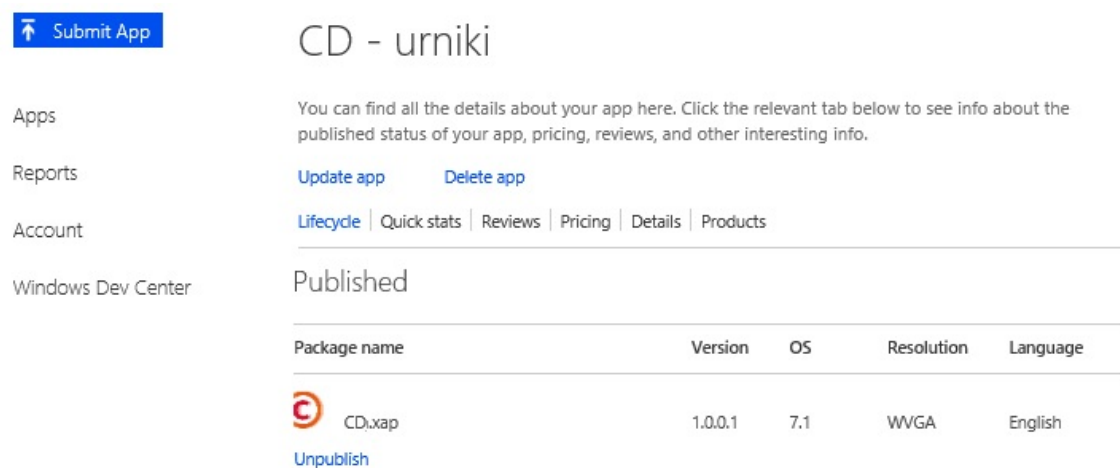
Windows Phone aplikacije se na mobilne aparate nameščajo preko trgovine imenovane Windows Store. Objava aplikacij je omogočena vsem uporabnikom oz. razvijalcem (ali podjetjem), ki se predtem registrirajo in poravnajo pristopnino. Za podjetja je potreben še postopek verifikacije.

Celotna končana aplikacija je zapakirana v datoteki s končnico XAP, ki jo v postopku objave aplikacije tudi uporabimo. Med celotnim postopkom objave je potrebno dodati še nekaj ključnih podatkov o sami aplikaciji:

- ime
- kategorija in podkategorija
- cena
- tržišča na katerih bo dostopna
- način distribucije
- opis aplikacije, ki bo viden uporabnikom
- ključne besede, po katerih je izvedeno iskanje
- ikone in slike različnih dimenzij

Po vnosu vseh zahtevanih podatkov in oddaji mora aplikacije še skozi Microsoftov sistem preverjanja, kar naj ne bi trajalo več kot pet delovnih dni. Aplikacija, ki uspešno prestane to preverjanje je potem dostopna preko trgovine vsem uporabnikom na območju, ki smo ga izbrali tekom objave.

Običajna objava aplikacije bi pomenila, da je le-ta dostopna vsem uporabnikom, vendar mi želimo omejiti namestitev in uporabo aplikacije samo na posamezne uporabnike, ki so naši zaposleni. Odločili smo se za objavo aplikacije kot »*beta*« pri čemer vpišemo uporabniška imena uporabnikov, ki bodo lahko dostopali do aplikacije in si jo namestili. Tem izbranim uporabnikom nato, po uspešnem preverjanju, pošljemo povezavo do aplikacije.



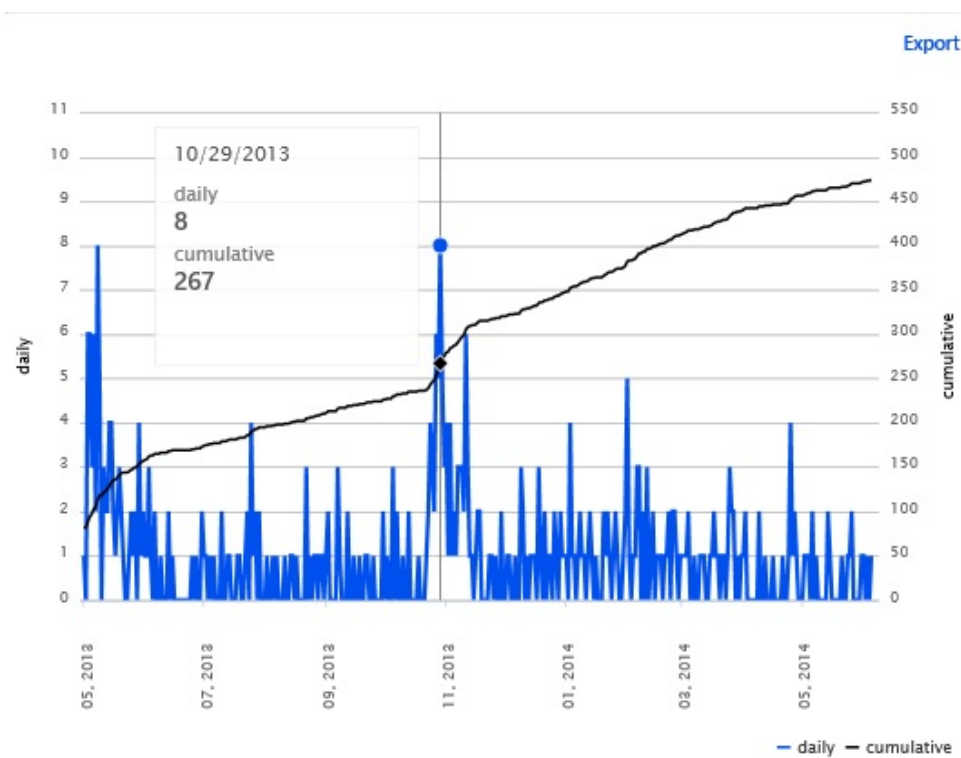
The screenshot shows the 'CD - urniki' page in the Windows Dev Center. On the left is a sidebar with links: 'Submit App', 'Apps', 'Reports', 'Account', and 'Windows Dev Center'. The main content area has a header 'CD - urniki' and a sub-header 'Published'. Below this is a table with columns: 'Package name', 'Version', 'OS', 'Resolution', and 'Language'. The table contains one entry for 'CD_xap' with version '1.0.0.1', OS '7.1', Resolution 'WVGA', and Language 'English'. There are also links for 'Update app', 'Delete app', 'Lifecycle', 'Quick stats', 'Reviews', 'Pricing', 'Details', 'Products', and 'Unpublish'.

Package name	Version	OS	Resolution	Language
CD_xap	1.0.0.1	7.1	WVGA	English

Slika 4.15: Objavljena aplikacija

Poleg tega načina je mogoče tudi odprtje svoje “privatne” trgovine, kar je zelo uporabno za podjetja, ki želijo aplikacije direktno prenesti na naprave svojih zaposlenih[15].

Podjetja, ki se želijo poslužiti tega načina distribucije, se morajo najprej registrirati in prejeti ustrezne certifikate. Temu sledi kreiranje svojega AET (Application Enrollment Token) žetona, kateri mora biti nato nameščen na vseh napravah, ki želijo dostopati do te trgovine.

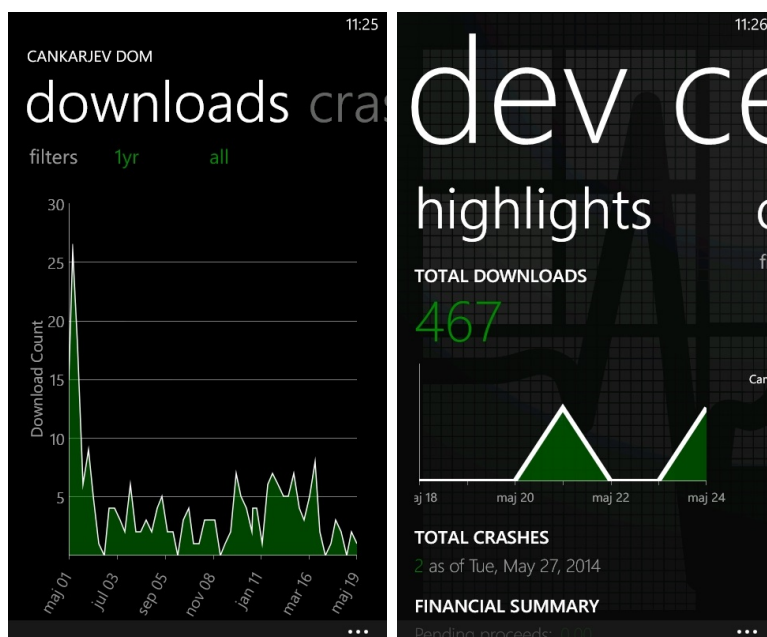


Slika 4.16: Graf iz poročila o nameščanju aplikacije

4.9.1 Nadzorna plošča trgovine

Objavljene aplikacije lahko spremljamo v nadzorni plošči sistema trgovine, kjer lahko med drugim tudi spremljamo število inštalacij, morebitne težave z aplikacijami, mnenja uporabnikov, finančna poročila in nenazadnje poskrbimo tudi za nadgradnje aplikacij.

Omogoča nam dostop do podrobnih podatkov o posameznih namestitvah naših aplikacij. Ta poročila vsebujejo podatke o posameznikovi državi, jeziku, ki ga uporablja na napravi, starostni skupini, ki ji pripada, spolu, proizvajalcu naprave, verziji OS in mobilnemu operaterju. Na osnovi teh poročil pa si lahko izdelamo profil najpogostejših uporabnikov naših aplikacij.

Slika 4.17: Aplikacija *DevCenter*

Za spremljanje naše trgovine obstaja tudi mobilna aplikacija *Dev Center* podjetja Microsoft, kjer imamo pravtako na pregled vse informacije o naših aplikacijah.

Poglavje 5

Ideje za nadaljevanje

Tekom razvoja aplikacije se vedno porodijo vedno nove in nove ideje, kaj bi še lahko dodali sami aplikaciji, vendar je velikokrat potrebno njihovo realizacijo, ker preveč posegajo v samo začetno zasnovo, prestaviti na kasnejše izdaje oz. nadgradnje aplikacije.

Pri tem je zelo pomemben tudi odziv samih uporabnikov aplikacije in v veliki meri je potrebno upoštevati tudi njihove želje. Nekaj idej za nadgradnjo naše aplikacije:

- kreiranje lokalne podatkovne zbirke, ki bi se uporabljala v primeru nedostopnosti do omrežja. V to lokalno podatkovno zbirko bi shranjevali kontaktne podatke zaposlenih, ki bi jih lahko kasneje na zahtevo osveževali.
- povezava s sistemom elektronskega shranjevanja pogodb, s čimer bi pooblaščenim uporabnikom omogočili dostop do pogodb, ki se nanašajo na posamezen projekt
- povezava s finančno-računovodskim sistemom, ki bi vodjem projektov omogočiti pregled nad finančnim stanjem projekta (pregled računov, statistika, stanje, ...)
- “LiveTile” s seznamom trenutnih zadolžitev zaposlenega

Slike

2.1	Primer prikaza urnikov v spletni aplikaciji	4
3.1	Primer začetnega zaslona	8
3.2	Visual Studio 2012	11
3.3	CLR [5]	13
3.4	Silverlight za WP - izbira tipov aplikacij v Visual Studiu	14
4.1	Podatkovni tok	18
4.2	Primer kontrolnega in podatkovnega toka v SSIS paketu	19
4.3	Začetni zaslon aplikacije	23
4.4	Gradnik <i>Panorama</i> [9]	24
4.5	Seznam urnikov in izbira dneva	26
4.6	Podrobno o urniku in seznam sodelujočih zaposlenih	27
4.7	Zgradba gradnika <i>Pivot</i>	28
4.8	Seznam projektov in iskalni kriteriji pri projektih	29
4.9	Podrobno o projektu	30
4.10	Prikaz zaposlenih	31
4.11	Prikaz kontaktnih podatkov	32
4.12	Primer izdelave ikone s programom Metro Studio 2.0	35
4.13	Izbira emulatorja in pomen povratnih informacij pri emulatorju	36
4.14	Emulator, menu in dodatna orodja	37
4.15	Objavljena aplikacija	39
4.16	Graf iz poročila o nameščanju aplikacije	40
4.17	Aplikacija <i>DevCenter</i>	41

Literatura

- [1] Poslovno poročilo CD za leto 2013, str. 17–21, 2014, dostopno na:
<http://www.cd-cc.si/media/PoslovnoPorociloCD2013.pdf>

- [2] Wikipedia o Windows Phone, dostopno na:
<http://en.wikipedia.org/wiki/WindowsPhone>

- [3] Co-development for Windows Phone 7/8 and Windows 8 guide, dostopno na:
<http://developer.nokia.com/community/wiki/Co-developmentforWindowsPhone7/8andWindows8guide>

- [4] Wikipedia o Visual Studiu, dostopno na:
<http://en.wikipedia.org/wiki/MicrosoftVisualStudio>

- [5] Wikipedia o .NET ogrodju, dostopno na:
<http://en.wikipedia.org/wiki/.NET-Framework>

- [6] Wikipedia o CLR (Common Language Runtime), dostopno na:
<http://en.wikipedia.org/wiki/Common-Language-Runtime>

- [7] Wikipedia o Microsoft Silverlight, dostopno na:
<http://en.wikipedia.org/wiki/Microsoft-Silverlight>

- [8] Wikipedia o SQL Server Integration Services, dostopno na:
<http://en.wikipedia.org/wiki/SQL-Server-Integration-Services>

- [9] MSDN: Panorama control for Windows Phone, dostopno na:
[http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941104\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941104(v=vs.105).aspx)
- [10] AdDuplex WP Statistics Report for April 2014, dostopno na:
<http://blog.adduplex.com/2014/04/adduplex-windows-phone-statistics.html>
- [11] Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013, dostopno na:
<http://www.gartner.com/newsroom/id/2665715>
- [12] MSDN: Pivot control architecture for Windows Phone 8, dostopno na:
[http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941097\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941097(v=vs.105).aspx)
- [13] MSDN: Saving and Loading Data, dostopno na:
[http://msdn.microsoft.com/en-us/library/gg680266\(v=pandp.11\).aspx](http://msdn.microsoft.com/en-us/library/gg680266(v=pandp.11).aspx)
- [14] MSDN: Windows Phone Emulator for Windows Phone 8, dostopno na:
[http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff402563\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff402563(v=vs.105).aspx)
- [15] MSDN: Company app distribution for Windows Phone, dostopno na:
[http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206943\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206943(v=vs.105).aspx)